IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: John X. Zhong, et al.

Assignee: Synopsys, Inc.

Title: DESIGN VERIFICATION BY SYMBOLIC SIMULATION USING A NATIVE HARDWARE DESCRIPTION LANGUAGE

Serial No.: 10/620,628 File Date: 07/15/2003

Examiner: Stacy Whitmore Art Unit: 2825

Docket No.: SYN-0551CON1 (formerly 4162P001C)

Date: May 10, 2006

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief, filed in triplicate, is in support of the Notice of Appeal dated March 16, 2006.

/

/

/

/

/

/

/

/

/

/

/

INDEX

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Synopsys, Inc., pursuant to the Assignments recorded in the U.S. Patent and Trademark Office on December 15, 2004 on Reel 015467, Frame 0940 and on November 22, 1999 on Reel 010421, Frame 0041.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no other appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-2, 4, 6-12, 14, and 16-20 are pending. Claims 3, 5, 13, and 15 are cancelled. Claims 1-2, 4, 6-12, 14, and 16-20 stand rejected.

In the present paper, rejected Claims 1-2, 4, 6-12, 14, and 16-20 are appealed.

Pending Claims 1-2, 4, 6-12, 14, and 16-20 are listed in VIII. Appendix A.

IV. STATUS OF AMENDMENTS

All claim amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In accordance with one aspect of the invention, the simulator can be instructed through the use of one or more statements (e.g. programming interface calls (PLIs)) that declare certain variables as symbols. Specification, page 11, lines 3-5. That is, a statement "may be used to indicate to the simulator that an object in a hardware description language used to indicate a signal (e.g. a Verilog object) is to be a symbol to the simulator, such that variable assigned objects are designated as symbolic variable objects". Specification, page 11, lines 5-9. "In this manner, the existing hardware description languages are able to support the specification of symbolic input". Specification, page 11, lines 9-11.

Figure 1 illustrates "a flow diagram of one embodiment of a process comprising a series of processing steps to simulate a design". Specification, page 12, lines 9-10. Processing block 101 (Figure 1) specifies to the simulator the signals to treat as symbolic. Specification, page 12, lines 13-14. "The signals may be inputs, interrupts, memory values, or any other portion of a design that may be represented as an object in a hardware description language". Specification, page 12, lines 14-16.

"In one embodiment, the process instructs the simulator using the following PLI command: $exp_var(a, b, …) which causes the simulator to designate the objects a, b, etc. as symbols." Specification, page 13, lines 5-8. "Upon encountering such a programming statement, the symbolic simulator propagates logic expressions, instead of binary values, capturing the relationship from input to output". Specification, page 11, lines 11-13.

Therefore, specifying a hardware description language object that represents at least one signal as a symbol in a

design using a first statement beneficially allows the existing hardware description languages to support the specification of symbolic input. As a result, a symbolic simulator can, in response to the first statement, treat the at least one hardware description language object as a symbol, thereby eliminating the need for the user to manually specify the signal during the simulation.

Appellants provide the following concise explanation of the subject matter defined in independent Claims 1 and 11.

Claim 1:

"specifying at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language" - refer to the Specification, page 11, lines 3-11, page 12, lines 13-16, page 13, lines 5-9, and FIG. 1, processing block 101.

"instructing a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol" - refer to the Specification, page 11, lines 3-11 and FIG. 1, processing block 101.

Claim 11:

"specify at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language" - refer to the Specification, page 11, lines 3-11, page 12, lines 13-16, page 13, lines 5-9, and FIG. 1, processing block 101.

"instruct a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol" - refer to the Specification, page 11, lines 3-11 and FIG. 1, processing block 101.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following issues are presented to the Board of Appeals for decision:

A. Whether Claims 1-2, 4, 6-8, 11-12, 14, and 16-18 are patentable under 35 U.S.C. 102(b) over "An Integral Environment For HDL Verification" (York).

B. Whether Claims 9-10 and 19-20 are patentable under 35 U.S.C. 103(a) over York in view of "The Verilog Procedural Interface For The Verilog Hardware Description Language" (Dawson).

VII. ARGUMENTS

A. Claims 1-2, 4, 6-8, 11-12, 14, and 16-18 are patentable under 35 U.S.C. 102(b) over "An Integral Environment For HDL Verification" (York).

1. York: Overview

York describes a formal verification prototype (Forte) that integrates formal verification and simulation. Abstract. Forte fits into a standard top-down design flow using standard HDLs, such as Verilog-HDL and VHDL. Abstract.

2. Appellants' limitations recited in Claims 1-2, 4, 6-8, 11-12, 14, and 16-18 are not taught by York.

Claim 1 recites:

> specifying at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language; and

instructing a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol.

"[S]pecifying ... [a] hardware description language object that represents at least one signal as a symbol in a design using a first statement" as recited in Claim 1, beneficially allows "the existing hardware description languages ... to support the specification of symbolic input." Specification, page 11, lines 9-11. As a result, "a symbolic simulator [can] in response to the first statement ... treat the at least one hardware description language object as a symbol" as recited by Claim 1, thereby eliminating the need for the user to manually specify the signal during the simulation.

The Examiner cites the following passages of York as teaching "specifying at least one hardware description language object that represents at least one signal as a symbol in a design" as recited by Claim 1: pgs. 12, 13, 14, Sections 3.1, 3.2, 3.3, and 3.4, and Figures 2 and 3. Specifically, the Examiner appears to focus on the examples of the ALU and UART (outputs and inputs, respectively) described by York.

Figure 2 (see below, for convenience) shows a simple Verilog example of an ALU and its corresponding intermediate format (IF), wherein high level operators are preserved. In this example of standard Verilog code for an ALU, a listing of various cases (modes of operations to be performed on the inputs a and b)(i.e. "out = a & b", "out = a | b", "out = a + b", and "out = a - b") are defined.

Verilog	IF
module alu (out, a, b, mode); output [4:0] out; input [3:0] a, b; input [1:0] mode; reg[4:0] out; always @(a or b or mode) case(mode[1:0]) 2'b00 : out = a & b; 2'b01 : out = a | b; 2'b10 : out = a + b; 2'b11 : out = a - b; endcase endmodule	MODULE alu (out, a, b, mode) INPUT mode : array 1 .. 0 of boolean; INPUT a : array 3 .. 0 of boolean; INPUT b : array 3 .. 0 of boolean; OUTPUT out : array 4 .. 0 of boolean; DEFINE out := ((mode = 0) ? (0 :: a & b) : ((mode = 1) ? (0 :: a | b) : ((mode = 2) ? a + b : a - b)));

Figure 2: Verilog and corresponding IF

Thus, the case choices in Figure 2 of York are explicit expressions of the output values for a given mode, and therefore do not "represent[] at least one signal as a symbol ... using a first statement that is part of a hardware description language" as recited by Claim 1.

Figure 3 of York shows a functional description of the receiver part of a UART. During normal operation, the data input "datain" is held low to indicate the start of the data byte and held high to indicate the end of the byte. An improper start or stop but resets the receiver. Thus, this data input to the UART also does not "represent[] at least one signal as a symbol ... using a first statement that is part of a hardware description language" as recited by Claim 1.

With respect to symbolic simulation, York explicitly teaches a methodology in which the user must "take the Verilog simulation, apply symbolic boolean variables to the inputs of the receivers in both descriptions, and simulate symbolically". York, page 14, section 3.3. Thus, in the methodology of York, symbolic inputs must be supplied at the start of the symbolic simulation. This methodology is distinct from instructing the

symbolic simulator in response to a first statement to treat the hardware description language object as a symbol.

Because York fails to describe or suggest "specifying at least one hardware description language object that represents at least one signal as a symbol" much less "instructing a symbolic simulator ... to treat the at least one hardware description language object as a symbol" (both limitations recited in Claim 1), Claim 1 is allowable under 35 U.S.C. 102(b) over York.

Claims 2, 4, 6, 7, and 8 depend from Claim 1, and are therefore allowable over York for at least the same reasons that Claim 1 is allowable. Accordingly, Appellants respectfully request reconsideration and allowance of Claims 1, 2, 4, 6, 7, and 8.

Claim 11 recites in part:

> executable instructions which ... cause the at least one processing device to:
>
> specify at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language; and
>
> instruct a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol.

Thus, for substantially the same reasons as described above with respect to Claim 1, Claim 11 is also allowable under 35 U.S.C. 102(b) over York.

Claims 12, 14, 16, 17, and 18 depend from Claim 11, and are therefore allowable over York for at least the same reasons that Claim 11 is allowable. Accordingly, Appellants respectfully request reconsideration and allowance of Claims 11, 12, 14, 16, 17, and 18.

B. Claims 9-10 and 19-20 are patentable under 35 U.S.C. 103(a) over York in view of "The Verilog Procedural Interface For The Verilog Hardware Description Language" (Dawson).

1. York: Overview (see Section A)

2. Dawson: Overview

Dawson describes a new C-programming interface (VPI) for the Verilog Hardware Description Language. Abstract. VPI provides a consistent object-oriented access to the complete Verilog HDL language. Abstract.

3. Appellants' limitations recited in Claims 9-10 and 19-20 are not taught by the combination of York and Dawson.

Dawson describes the Verilog Procedural Interface, which is a new C-programming interface for the Verilog Hardware Description Language. Abstract. However, Dawson fails to disclose or suggest the recited first statement and its use in instructing a symbolic simulator to treat a hardware description language object as a symbol. Therefore, Dawson does not remedy the deficiencies of York described above with respect to Claims 1 and 11.

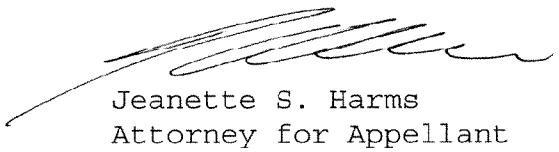
Claims 9-10 depend from Claim 1, whereas Claims 19-20 depend from Claim 11. Therefore, Claims 9-10 and 19-20 are allowable under 35 U.S.C. 103(a) over York in view of Dawson.

Accordingly, Appellants respectfully request reconsideration and allowance of Claims 9-10 and 19-20.

C. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of Claims 1-2, 4, 6-12, 14, and 16-20 are erroneous, and reversal of these rejections is respectfully requested.

Respectfully submitted,



Customer No.: 35273

Jeanette S. Harms
Attorney for Appellant
Reg. No. 35,537

Telephone: 408-451-5907
Facsimile: 408-451-5908

VIII. CLAIMS APPENDIX

Claim 1 (previously presented): A method for performing design verification, the method comprising:

specifying at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language; and

instructing a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol.

Claim 2 (previously presented): The method defined in Claim 1 further comprising:

inserting the first statement into a design specification; and

inputting the design specification into the symbolic simulator.

Claim 3 (cancelled)

Claim 4 (previously presented): The method defined in Claim 1 wherein the at least one hardware description language object comprises a Verilog object.

Claim 5 (cancelled)

Claim 6 (original): The method defined in Claim 1 wherein the at least one signal comprises an input.

Claim 7 (previously presented): The method defined in Claim 1 further comprising:

specifying a check using a second statement that is part of the hardware description language, the check to perform a test to validate design functionality; and

instructing the symbolic simulator using the second statement to perform the test.

Claim 8 (previously presented): The method defined in Claim 7 further comprising:

inserting the first and second statements into a design specification; and

inputting the design specification into the symbolic simulator.

Claim 9 (previously presented): The method defined in Claim 7 wherein the second statement comprises a PLI.

Claim 10 (original): The method defined in Claim 7 further comprising:

instructing the symbolic simulator to generate a file with information to locate an identified fault.

Claim 11 (previously presented): An article of manufacture having at least one recordable medium having stored thereon executable instructions which, when executed by at least one processing device, cause the at least one processing device to:

specify at least one hardware description language object that represents at least one signal as a symbol in a design using a first statement that is part of a hardware description language; and

instruct a symbolic simulator in response to the first statement to treat the at least one hardware description language object as a symbol.

Claim 12 (previously presented): The article of manufacture defined in Claim 11 further comprising executable instructions stored on the at least one recordable medium which, when executed by at least one processing device, cause the at least one processing device to:

insert the first statement into a design specification; and

input the design specification into the symbolic simulator.

Claim 13 (cancelled)

Claim 14 (previously presented): The article of manufacture defined in Claim 11 wherein the at least one hardware description language object comprises a Verilog object.

Claim 15 (cancelled)

Claim 16 (original): The article of manufacture defined in Claim 11 wherein the at least one signal comprises an input.

Claim 17 (previously presented): The article of manufacture defined in Claim 11 further comprising executable instructions stored on the at least one recordable medium which, when executed by at least one processing device, cause the at least one processing device to:

specify a check using a second statement that is part of the hardware description language, the check to perform a test to validate design functionality; and

instruct the symbolic simulator using the second statement to perform the test.

Claim 18 (previously presented): The article of manufacture defined in Claim 17 further comprising executable instructions stored on the at least one recordable medium which, when executed by at least one processing device, cause the at least one processing device to:

insert the first and second statements into a design specification; and

input the design specification into the symbolic simulator.

Claim 19 (previously presented): The article of manufacture defined in Claim 17 wherein the second statement comprises a PLI.

Claim 20 (original): The article of manufacture defined in Claim 17 further comprising executable instructions stored on the at least one recordable medium which, when executed by at least one processing device, cause the at least one processing device to:

instruct the symbolic simulator to generate a file with information to locate an identified fault.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None